(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2003－208306
（P2003－208306A）
(43)公開日　平成15年７月25日(2003.7.25)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)
Ｇ０６Ｆ	9/30	３１０	Ｇ０６Ｆ	9/30	３１０Ｃ　５Ｂ０１３
		３５０			３５０Ａ　５Ｂ０３３
					３５０Ｆ　５Ｂ０８１
	9/38	３７０		9/38	３７０Ｘ
	9/45			9/44	３２２Ａ

審査請求　未請求　請求項の数11　ＯＬ　（全７頁）

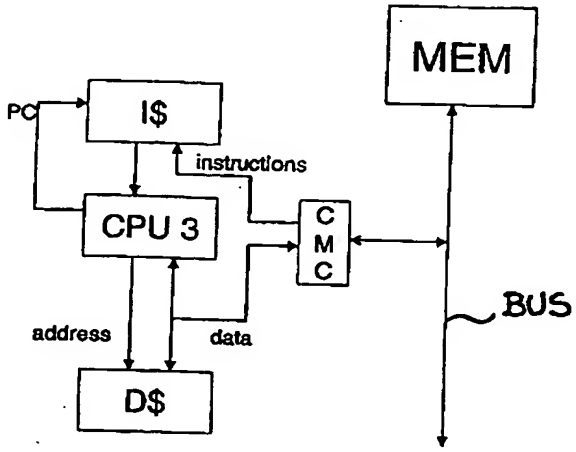
(21)出願番号　特願2002－347918(P2002－347918)
(22)出願日　平成14年11月29日(2002.11.29)
(31)優先権主張番号　０１８３０８１４：８
(32)優先日　平成13年12月27日(2001.12.27)
(33)優先権主張国　欧州特許庁（ＥＰ）

(71)出願人　591002692
エスティーマイクロエレクトロニクスエス．アール．エル．
ＳＴＭｉｃｒｏｅｌｅｃｔｒｏｎｉｃｓ Ｓ．ｒ．ｌ．
イタリア国　ミラノ　20041　アグラーテ　ブリアンツァ　ヴィア　ツィー　オリヴェッティ　２
(74)代理人　100072051
弁理士　杉村　興作　（外１名）

最終頁に続く

(54)【発明の名称】　プロセシングアーキテクチャ、該アーキテクチャ実現システム及び該システムのオペレーション方法

(57)【要約】　（修正有）
【課題】　２以上の異なる命令セットを実行し得るプロセシングアーキテクチャの提供。
【解決手段】　このアーキテクチャは、第１セットの命令（OsTask1.1，OsTask1.2，．．．）と第２セットの命令（MmTask2.1，MmTask2.2，MmTask2.3，．．．）の両方を実行するよう構成された単一ＣＰＵを具える。該単一ＣＰＵは、該ＣＰＵが前記第１セットの命令（OsTask1.1，OsTask1.2，．．．）を実行する第１動作モードと該ＣＰＵが前記第２セットの命令（MmTask2.1，MmTask2.2，MmTask2.3，．．．）を実行する第２動作モードとの間で切り換えるように構成されている。この解決方法は種々のＣＰＵに対する２つ以上の実行モード間で複数の切り換え命令を用いることにより一般化することができる。



【特許請求の範囲】

【請求項１】　第１のＣＰＵ（ＣＰＵ１）により実行されるようにコンパイルされ第２のＣＰＵ（ＣＰＵ２）では実行できない少なくとも一つの第１の命令セット（OsTask1.1, OsTask1.2, ．．．）と第２のＣＰＵ（ＣＰＵ２）により実行されるようにコンパイルされ第１のＣＰＵ（ＣＰＵ１）では実行できない少なくとも一つの第２の命令セット（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）を実行するプロセシングアーキテクチャであって、該アーキテクチャは、前記第１セットの命令（OsTask1.1, OsTask1.2, ．．．）と前記第２セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）の両方を実行するよう構成された単一プロセッサ（ＣＰＵ３）を具え、該単一プロセッサ（ＣＰＵ３）は、該プロセッサが少なくとも前記第１セットの命令（OsTask1.1, OsTask1.2, ．．．）を実行する第１動作モードと該プロセッサが前記第２セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）を実行する第２動作モードとの間で選択的に切り換え可能であり、前記単一プロセッサ（ＣＰＵ３）は、少なくとも前記第１動作モードと前記第２動作モードとの間で少なくとも一つの切り換え命令（Mode1_NotMode2 flag）を認識し、前記少なくとも一つの切り換え命令に従って前記第１動作モードと前記第２動作モードとの間の切り換えを実行するよう構成することを特徴とするプロセシングアーキテクチャ。

【請求項２】　前記単一プロセッサ（ＣＰＵ３）はデータ用の単一キャッシュ（ＤＳ）に関連することを特徴とする請求項１記載のアーキテクチャ。

【請求項３】　前記単一プロセッサ（ＣＰＵ３）は命令用の単一キャッシュに関連することを特徴とする請求項１又は２記載のアーキテクチャ。

【請求項４】　前記単一プロセッサ（ＣＰＵ３）はバスを介してメインメモリと対話する単一インタフェース（ＣＭＣ）に関連することを特徴とする請求項１－３の何れかに記載のアーキテクチャ。

【請求項５】　メモリ内の前記命令をアドレスするために単一プログラムカウンタが設けられていることを特徴とする請求項１－４の何れかに記載のアーキテクチャ。

【請求項６】　前記単一プログラム（ＣＰＵ３）は、前記第１セットの命令（OsTask1.1, OsTask1.2, ．．．）及び前記第２セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）をそれぞれデコーディングする少なくとも一つの第１のデコーディングモジュール（ＤＥＣ１）及び少なくとも一つの第２のデコーディングモジュール（ＤＥＣ２）を具えることを特徴とする請求項１－５の何れかに記載のアーキテクチャ。

【請求項７】　前記第１セットの命令（OsTask1.1, OsTask1.2, ．．．）及び前記第２セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）のオペランドを読み出すためのレジスタの統合ファイル（レジスタファイル）を具えることを特徴とする請求項１－６の何れかに記載のアーキテクチャ。

【請求項８】　前記第１の動作モード又は前記第２の動作モードにおいて命令の実行に使用されないとき選択的に非活性化し得るユニットを具えることを特徴とする請求項１－７の何れかに記載のアーキテクチャ。

【請求項９】　請求項１－８の何れかに記載のアーキテクチャを実現するプロセシングシステム。

【請求項１０】　請求項９に記載のプロセシングシステムを使用する方法であって、

前記少なくとも一つの第１の命令セット（OsTask1.1, OsTask1.2, ．．．）の命令のセットと前記少なくとも一つの第２の命令セット（MmTask2.1, MmTask2.2, MmTask2.3, ．．．）の命令のセットをコンパイルするオペレーションと、

前記少なくとも一つの切り換え命令（Mode1_NotMode2 flag）を前記命令のセットの先頭に付与するオペレーションと、を具えることを特徴とするプロセシングシステムの使用方法。

【請求項１１】　前記第１のＣＰＵ（ＣＰＵ１）及び前記第２のＣＰＵ（ＣＰＵ２）のコンパイルフローをそのまま用いて各プロセスをコンパイルし、前記切り換え命令（Mode1_NotMode2 flag）を前記命令のセットの先頭に入力することを特徴とする請求項１０記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、プロセシングアーキテクチャ及び該アーキテクチャを実現するシステムに関するものである。本発明は特に移動通信システムに使用し得るマイクロプロセシングアーキテクチャに注目して開発したものである。しかし、本発明の範囲はこの応用分野に限定されるものではない。

【０００２】

【従来技術】セルフォンの代表的なシステムアーキテクチャは複数の中央処理装置（ＣＰＵ）のアベイラビリティに基づいている。これらのＣＰＵは通常２つであり、各々特定の目的を満たす。第１のＣＰＵはオペレーティングシステムの制御機能に実質的に類似する制御機能を実行する。このタイプの用途は計算上の観点からの厳しい要求も、高い性能も要求しない。通常、簡単なフェッチーデコードーリードー実行ーライトバックのステージからなるスケーラパイプライン形のアーキテクチャの使用を想定している。第２のＣＰＵは計算上の約束及び性能に関して相違する特性を有する機能を実行する。この理由のために、通常、スーパスケーラプロセッサ又は１サイクルにつき複数の命令を発行し実行することができるベリーロングインストラクションワード（ＶＬＩＷ）パイプラインプロセッサの使用を想定している。これらの命令は（ＶＬＩＷアーキテクチャの場合には）コンパイルステージで、（スーパスケーラプロセッサの場合に

は）実行ステージでスケジューリングすることができる。

【０００３】この計算リソースの重複はメモリに関する要件の重複をまねき、その結果として大きな電力消費を生ずる。大きな電力消費は２つのＣＰＵの一方又は他方を交互にスリープモードに設定することにより部分的に制限できるが、避けることはできない。

【０００４】図１につき説明すると、上述したタイプのワイヤレス応用に対する代表的なアーキテクチャはＣＰＵ１及びＣＰＵ２で示す２つのマイクロプロセッサのような２つのＣＰＵを具え、各自キャッシュメモリアーキテクチャを具えている。

【０００５】ＣＰＵ１は代表的には３２ビットパイプラインスケーラマイクロプロセッサである。これは、その内部アーキテクチャが種々のロジックステージからなり、各ロジックステージが命令を一つの特定の状態で含むことを意味する。この状態は下記の状態：

－メモリからの命令のローディング；

－デコーディング；

－レジスタファイルのアドレッシング；

－実行；及び

－メモリからのデータの書込み/読み出し；

の一つである。ビット数はＣＰＵ１が動作するデータ及び命令の長さに関連する。命令はコンパイルにより特定の順序で発生され、その順序で実行される。

【０００６】ＣＰＵ２は代表的には１２８ビットパイプラインスーパスケーラマイクロプロセッサ又はＶＬＩＷマイクロプロセッサである。これは、その内部アーキテクチャが種々のロジックステージからなり、いくつかのロジックステージが、例えば実行ステップで、命令を並列に実行することができることを意味する。代表的には、４つの３２ビット命令（１２８ビットに相当する）を並列に実行し、データは３２ビットで表現する。並列に実行可能な複数の実行ステージを供給するために命令を実行中に動的に順序付ける場合及び命令が相互に依存せず、従ってソースコードのコンパイルにより静的に発生される順序を変更する場合には、そのプロセッサはスーパスケーラであるという。

【０００７】その代わりに、命令をコンパイルステップにおいて静的に順序付けし、固定の順序で実行する（実行中に変更不能）場合には、そのプロセッサはＶＬＩＷ（ベリーロングインストラクションワード）という解決手法に対応する。

【０００８】再び図１に付き説明すると、各プロセッサＣＰＵ１，ＣＰＵ２は各自ＤＳで示すデータキャッシュとＩＳで示す命令キャッシュを有するため、メインメモリＭＥＭから処理すべきデータと実行すべき命令を並列にロードすることができる。２つのプロセッサＣＰＵ１，ＣＰＵ２はシステムバスにより相互接続されるとともに、このシステムバスによりメインメモリＭＥＭに接

続される。２つのプロセッサＣＰＵ１，ＣＰＵ２はバスへのアクセスを競争する。このアクセスは、これらのプロセッサが実行すべき命令又はデータ又はその両方がメインメモリ内に位置し、各自のキャッシュにないときに、コアメモリコントローラＣＭＣというそれぞれのインタフェースを介して達成される。このようなシステムは２つのマイクロプロセッサを使用し、それぞれに対応する２つのメモリ階層が不可欠であり、占有面積及び電力消費の点から幾分費用がかかることが理解される。一例として、代表的な応用では、ＣＰＵ１は通常１６キロバイトのデータキャッシュと１６キロバイトの命令キャッシュを有し、ＣＰＵ２は３２キロバイトのデータキャッシュと３２キロバイトの命令キャッシュを有する。

【０００９】図２はＣＰＵ１のロジックスキームを示す。第１ステップは実行すべき命令が関連する命令キャッシュＩＳのメモリアドレスを発生する。このアドレス（プログラムカウンタという）は命令のローディング（Fetch）を生じさせ、フェッチした命令をデコードし（Decode）、オペランドをアドレスするビットフィールドから機能（例えば、レジスタファイルに位置する２つのレジスタ内の２つの値の加算）を定義するビットフィールドを分離する。オペランドアドレスはレジスタファイルに送られ、これから命令のオペランドを読み出す。オペランドと実行すべき命令を定義するビットは実行ユニット（Execute）に送られ、実行ユニットが所望の演算（例えば加算）を実行する。次いでその結果はレジスタファイル内のメモリにリストアされる（Writeback）。

【００１０】その代わりに、ロード/ストアユニットが、メモリデータの読出し/書込みをこの目的専用の特定の命令を用いて行うことができる。他方、命令セットと（マイクロ）プロセシングアーキテクチャとの間に一対一の一義的対応(biunivocal)が存在することが容易に認識される。

【００１１】ＣＰＵ１について述べたことはＣＰＵ２にもほぼあてはまり、図３に同様の表現で示されている。主要な違いは、ＣＰＵ２の場合には、スーパスケーラ及びＶＬＩＷプロセッサで並列に動作し得る使用可能な実行ユニットが多数ある点にある。この点に関し、図３にはExcute2.1, Excute2.2, . . . , Excute2.nにより示された種々のステージが示されている。しかし、この場合にも、命令セットとプロセシングアーキテクチャとの間に一対一の一義的対応が存在する。例えばワイヤレスプロセッサのアーキテクチャのようなアーキテクチャでは、２つの命令セットが相違することが一般に見られる。このことは、ＣＰＵ１で実行される命令をＣＰＵ２で実行できず、その逆も同様であることを意味する。

【００１２】図４及び図５に関して、このような２つの各別の命令セットの形を取る複数のタイプのプロセシスを処理する必要があるものとする。例えば、先に述べた

5

応用コンテキスト（移動通信）に関して、２つのタイプのプロセス：

－ＣＰＵ１で実行されるオペレーティングシステムプロセスに類似するプロセスOsTask1.2, OsTask1.2, 等；及び

－ＣＰＵ２で実行されるコンテンツ（通常オーディオ/ビデオ/グラフィックコンテンツのようなマルチメディアコンテンツ）のプロセシングに関するプロセスMmTask2.1, MmTask2.2, MmTask2.3, 等；

に区別することができる。

【００１３】前者のプロセスはＣＰＵ１のコンパイラにより発生される命令を含み、従ってＣＰＵ１自身で実行することができるがＭＣＰＵ２で実行することはできない。第２のプロセスに対しては、正確にその逆が当てはまる。更に、各ＣＰＵは各自のコンパイルフローで特徴付けられ、各ＣＰＵのコンパイルフローは他の使用ＣＰＵのコンパイルフローと独立である点に注意されたい。

【００１４】図５は、上記のタスクのスケジューリングのシーケンスを２つのプロセッサＣＰＵ１及びＣＰＵ２にどのように分配するかを示す。上記のプロセスの総実行時間を１００とした場合、前者が総実行時間の１０％を占め、後者が９０％を占めるのが代表的である。これから、ＣＰＵ１は時間の１０％しか動作しないので、ＣＰＵ１は時間の９０％が冗長となることになる。

【００１５】上述の特性を利用してＣＰＵ１をターンオフすることによりエネルギーの節約を達成することができる。しかし、パワーダウンプロシージャはプロセシングの追加の待ち時間を導入し、この時間が上述の１０％に付加される。これらのプロシージャは実際には、

－レジスタファイルの全ての内部レジスタ並びにコア内に存在する他のユニット（例えばデコーディングユニット、実行ユニット）に供給するクロックをゲートすることによりレジスタファイルを除くＣＰＵのパワーダウン；

－ＣＰＵの完全なパワーダウン、キャッシュメモリへのエネルギー供給は維持；

－ＣＰＵ全体並びにデータキャッシュ及び命令キャッシュのパワーダウン；

を含む。

【００１６】構造上の点から見ると、上述したオペレーションに続いてプロセッサのパワーを再投入する時にパワーダウン前のプロセッサ自体を特徴づけるプロセッサの状態を復元する必要があるため、導入される待ち時間は数十ナノ秒から数十/数百ミリ秒の範囲になる。従って、上述のパワーダウンプロシージャはエネルギーの点からも計算の点からも効果的でない。

【００１７】

【発明が解決しようとする課題】従って、本発明の目的は、上述した欠点を克服することができるマイクロプロセシングシステムアーキテクチャを構成することにある。

6

【００１８】

【課題を解決するための手段】本発明によれば、この目的は特許請求の範囲において特定した特徴を有するアーキテクチャによって達成することができる。本発明は対応するシステム、並びに対応する使用方法にも関する。

【００１９】本発明の解決方法は、本質的には、上述した態様に従うオペレーティングに予想される制御コードをサポートするのに必要とされるリソース（ＣＰＵ、メモリ等）の二重化、一般に多重化は、最初から想定されている２つ（又はそれ以上）のＣＰＵを単一の最適化された（マイクロ）アーキテクチャ、即ち種々のＣＰＵのコンパイラにより発生された命令を実行し得る単一の新しいプロセッサに融合させることができれば避けることができるという認識に基づくものである。この場合には、前記新プロセッサは１以上の特別命令をデコードし、例えばその機能を異なる命令セットに固有の２以上の実行モードの間で切り換えることができなければならないという唯一の要件がある。

【００２０】この命令又はこれらの命令はＣＰＵに予め関連するコンパイラを用いてコンパイルされた命令の各セットの先頭に入れる。特に２つのステップが想定される。第１のステップはＣＰＵ１又はＣＰＵ２のコンパイルフローを（そのまま）用いて各プロセスをコンパイルする（以後、本発明は任意の数のこのようなＣＰＵに適用し得るが、説明を簡単にするために２つの出発ＣＰＵについて説明する）。第２のステップは各命令セットを取り、その先頭に特別命令を入力し、最適化されたマイクロアーキテクチャのフレームワークにおけるＣＰＵ１の実行モードとＣＰＵ２の実行モードとの間のモード切り換えを信号し許可する。

【００２１】以上の解決方法はメモリ及び電力消費に関し著しい節約をもたらす。更に、切り換え命令を検出する１つのフェッチユニット、（２つのＣＰＵ、ＣＰＵ１及びＣＰＵ２の各々に対する）２つのデコーディングユニット、単一のレジスタファイル、複数の実行ユニット、及びロード/ストアユニット（特別命令が検出されると、構成される）を使用するのみとすることができる。

【００２２】

【発明の実施の形態】本発明を図面を参照して実施例につき詳細に説明するが、本発明はこのような実施例に限定されるものではない。

【００２３】本発明の模範的な実施例の詳細な説明

上述したように、本発明が基づく主な思想は、低い計算ウェイト（例えば時間の１０％）のプロセスの実行をサポートするために、プロセシングリソースの二重化は不要であるという認識にある。図６に図式的に示すように、本発明の解決方法は、既知の解決方法ではＣＰＵ１及びＣＰＵ２のような２以上の個別のＣＰＵで実行するよう

設計されたプロセスをアプリケーションによらずに実行し得るＣＰＵ３で示す新しいプロセッサ又はＣＰＵアーキテクチャを定義するものであり、新しいアーキテクチャに対しリコンパイルする必要がある。

【００２４】基本的には、本発明の解決方法のねらいは、各ＣＰＵに想定されたもとのコンパイルフローを再利用し、その下流に、対応するプロセスの実行をコンパチブルにする第２ステップを付加することにある。

【００２５】特に、図７を参照して、第１のコンパイルステップにおいて、オペレーティングシステム用のプロセスOsTask1.1のソースコードを考察する。図１に示すようの伝統的なアーキテクチャでは、対応する命令は対応するコンパイラを用いてＣＰＵ１で実行する必要がある。次に、同じ第１ステップにおいて、マルチメディアオーディオ/ビデオ/グラフィックアプリケーション用のプロセス(MmTask2.1)のソースコードのコンパイルを考察する。このプロセスは、図１に示すような伝統的なアーキテクチャでは、この場合にもＣＰＵ１のコンパイラと異なる対応するコンパイラを用いてＣＰＵ２で実行される。更に、図１に示すようなスキームでは、２つのプロセッサＣＰＵ１及びＣＰＵ２が独立の命令セットのアーキテクチャを有する点を思い出す必要がある。

【００２６】次に、第２のステップを考察する。このステップでは（少なくとも）一つの新しい特別命令を丁度発生された命令の頭に入力する。この特別命令は対応する命令セットに続く命令の所属を識別することができる。従って、この特別命令はＣＰＵ３がＣＰＵ１の命令セットの実行モードからＣＰＵ２の命令セットの実行モードに、及びその逆に切り換わることができるようにする手段を表す。

【００２７】図８は、図１のアーキテクチャを、ＣＰＵ３で示す単一のＣＰＵに、関連するキャッシュメモリ、即ちデータキャッシュメモリＤＳ及び命令キャッシュメモリＩＳを設けることによりどのように巨視的観点から簡略化できるかを示す。このように、対応するメモリサブシステムはキャッシュメモリの二重化を必要とせず、対応するバスにインタフェースするインタフェースＣＭＣを経るメインメモリＭＥＭへのアクセス要求の競合から開放される。これから性能の明らかな改善が得られる。

【００２８】他方、プロセッサＣＰＵは、対応するコンパイラにより発生されＣＰＵ１のタイプのプロセッサで実行すべき命令もＣＰＵ２のタイプのプロセッサで実行すべき命令も実行できなければならない。これは同様に、２つのＣＰＵ間の実行モードの制御命令も実行できることを想定している。

【００２９】図９はここに提案するＣＰＵ３のロジックスキームを示す。命令はメモリにて単一プログラムカウンタによりアドレスされ、Fetch & Alignで示すユニットによりロードされる。このユニットは次に命令をＣＰＵ１及びＣＰＵ２の命令セットにコンパチブルなデコーディングユニットに送る。これらのデコーディングユニットは両方とも、命令セット１に対する実行モードを命令セット２に対する実行モードに及びその逆に切り換える特別命令の存在を検出することができる。こうして活性化されたフラグをＣＰＵ内に存在する全てのユニットに送って、そのＣＰＵ１－コンパチブルモード又はＣＰＵ２－コンパチブルモードのオペレーションを構成する。特に、図９において、このフラグはMode1_NotMode2 flagとして示す信号と同一である。最も簡単な実施例では、このフラグは、ＣＰＵ３がＣＰＵ１の命令セットで動作するとき論理値“１”を有し、ＣＰＵ３がＣＰＵ２の命令セットで動作するとき論理値“０”を有する。これと反対にすることもできること勿論である。

【００３０】ロードされた後続の命令を（Dec１及びDec2で示すステージで）デコードし、機能(例えば加算)を定義するビットフィールドをオペランドをアドレスするビットフィールドから分離する。対応するアドレスをレジスタファイルに送り、レジスタファイルから命令のオペランドを読み出す。これらのオペランドと実行すべき機能を定義するビットを要求されたオペレーションを実行する多数の実行ユニット（Execute1, . . . , Executem, Execute2.2, Executem+1, . . . , Executen, Execute . . . ）に送る。次に、結果を図２及び図３と同様にライトバックステージでレジスタファイルにリストアすることができる。その代わりに、ロード/ストアユニットがメモリから/メモリへのデータの読出し/書込みを行うことができ、各オペレーティングモードにはこの目的用の命令が存在する。

【００３１】特に、現在使用されていない実行モードとコンパチブルであるユニット（例えば、デコーディングユニットDec1及びDec2）は電力を消費しないように適切に“ターンオフ”することができることが認識されるであろう。

【００３２】本発明の原理を損なうことなく、上述した構成及び実施例の細部は特許請求の範囲に限定された本発明の範囲から逸脱することなく広範囲に変更することができること勿論であり、特に本発明の解決方法は種々のＣＰＵに対する２以上の実行モードの間で複数の切り換え命令を用いることにより一般化することができること明らかである。

【図面の簡単な説明】

【図１】　２以上のＣＰＵを用いる従来のプロセシングシステムの構成図である。

【図２】　図１の従来のプロセシングシステムのＣＰＵ１のロジックスキームを示す図である。

【図３】　図１の従来のプロセシングシステムのＣＰＵ２のロジックスキームを示す図である。

【図４】　ＣＰＵ１で実行されるタスクとＣＰＵ２で実行されるタスクの実行時間の比較を示す図である。

【図５】　図４のタスクが２つのプロセッサＣＰＵ１及びＣＰＵ２にどのように分配されるかを示す図である。

【図６】　本発明によるＣＰＵアーキテクチャを示す概念図である。

【図７】　本発明によるアーキテクチャにおけるコンパイルステップを示す図である。

【図８】　本発明によるアーキテクチャをブロック図で示したものである。

【図９】　図８に示すアーキテクチャの構造の詳細とオペレーションの詳細を示す図である。

【符号の説明】

ＣＰＵ３　単一ＣＰＵ

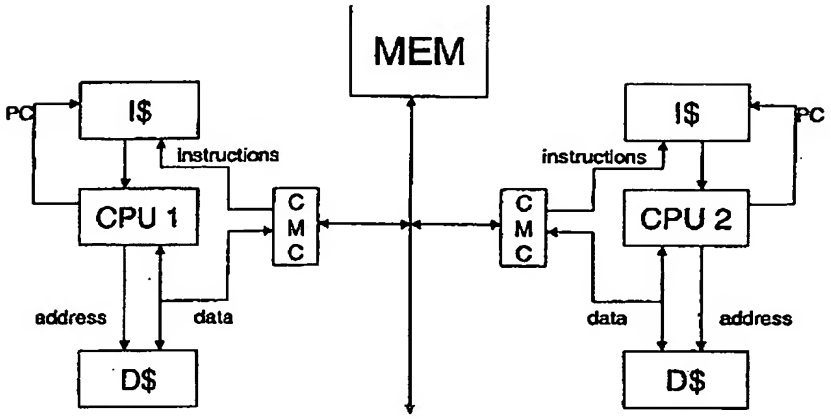
ＩＳ　命令キャッシュメモリ

ＤＳ　データキャッシュメモリ

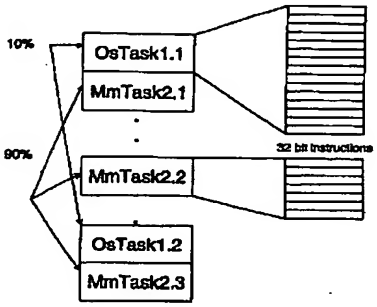
ＭＥＭ　メインメモリ

ＣＭＣ　インタフェース

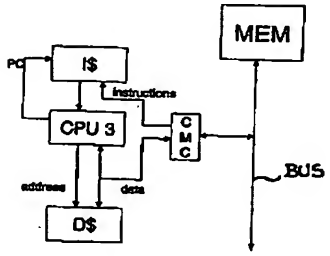
【図１】



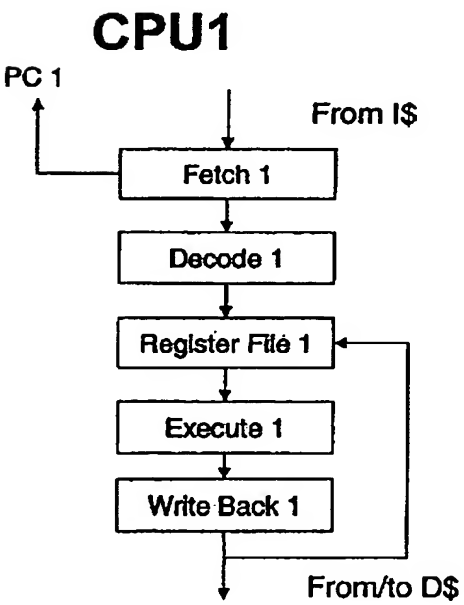
【図４】



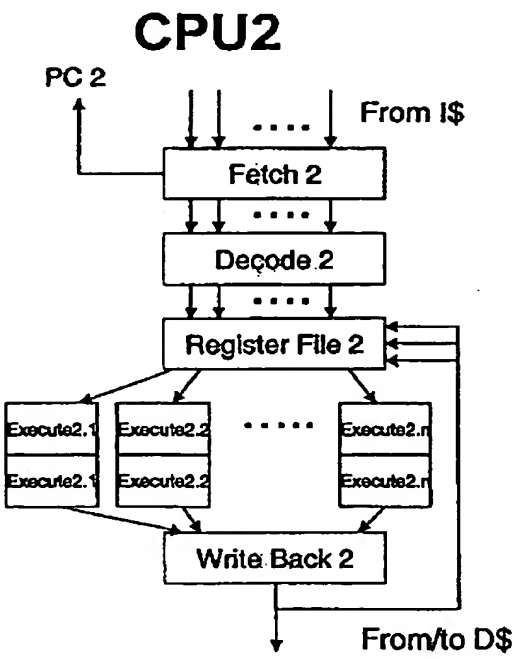
【図８】



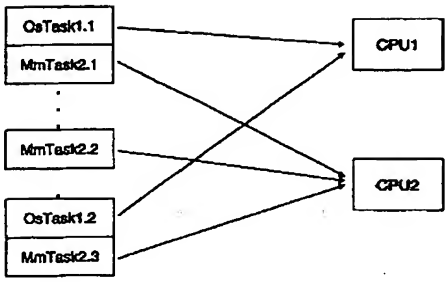
【図２】



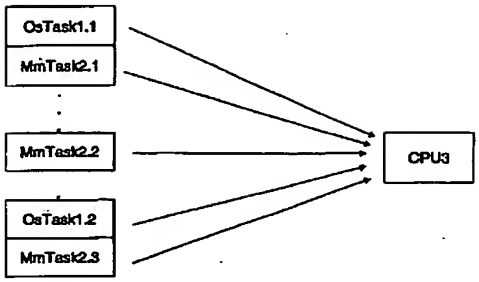
【図３】



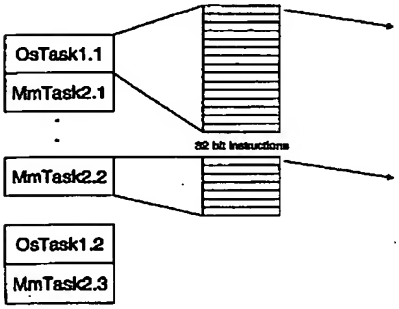
【図５】



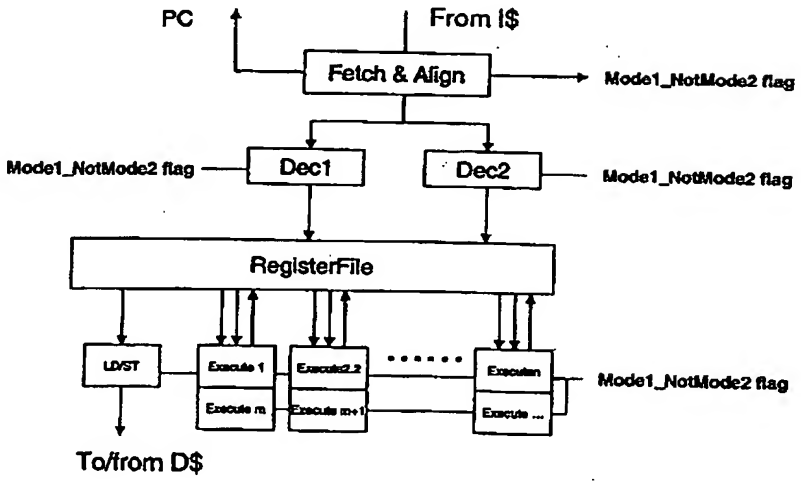
【図６】



【図７】



【図９】



フロントページの続き

(72)発明者 アレッサンドロ クレモネージ
イタリア国 ロディ 20079 エッセ アンジェロ ロディジアーノ ヴィア ダイツ 102
(72)発明者 ファブリツィオ ロヴァティ
イタリア国 ミラノ 20092 チニゼッロ バルサモ ヴィア コッレオーニ 15
(72)発明者 ダニーロ パウ
イタリア国 ミラノ 20099 セスト サン ジョヴァンニ ヴィア ダンテ 131

Fターム(参考) 5B013 DD04
5B033 AA05 AA10 AA14 AA15 BA05 BE00 BE05
5B081 CC00 CC53